Amendments to the Claims

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (Currently amended) A server method for converting Java-based objects into JavaScript-based objects, the method comprising:

a) identifying one or more Java-based object classes;

b) determining instance data based on the Java-based object classes;

c) introspecting each Java-based object class;

d) automatically creating an artifact representing a software model describing the instance data~~, the artifact mirroring the Java-based object hierarchy~~;

e) generating at least one converter ~~one or more converters~~ for each Java-based object ~~class~~, ~~each~~ the at least one converter being based on the artifact and configured for receiving the instance data and generating JavaScript code for converting each Java-based object to a JavaScript object and~~for recreating an instance from the classes as JavaScript-based objects,~~ for projecting ~~display~~ the JavaScript objects on a browser web page~~client of the server~~;

wherein the generating step comprises: ~~converters together form a compiled recursive descent parser, synching the object hierarchies by~~

maintaining rules that enforce typing and other ~~from the instance data that enforce Java~~ modeling constraints not found in JavaScript; and

defining a smallest number of properties of each of the JavaScript objects that together represent its uniqueness to produce a string that is portable across any environment and represents each JavaScript object's uniqueness, such that if there are two physical copies of what is logically the same JavaScript object, the method automatically aliases them out such that in the browser, there is only one copy of each duplicated JavaScript object;

f) generating a simple wrapper for invoking the at least one converter

~~f) invoking the converters at runtime to directly generate JavaScript code for projecting~~

~~onto the browser client~~.

2. (Canceled)

3. (Previously presented) The method of claim 1 wherein the object classes comprise JavaBeans and wherein the converters are invoked at runtime to generate JavaScript code required to recreate instance data from the JavaBeans into JavaScript objects in the browser.

4. (Previously presented) The method of claim 3 wherein determining, for each property in a class, comprises determining whether the property is an attribute.

5. (Previously presented) The method of claim 4 wherein invoking comprises outputting a JavaScript code for the value of the attribute.

6. (Previously presented) The method of claim 4 further comprising wherein creating the artifact comprises creating a buffer to hold an exported object value or values if the property is an array.

7. (Original) The method of claim 6 further comprising creating an array to hold object IDs for the referenced objects for each object in the array.

8. (Original) The method of claim 7 further comprising computing a signature of the object if the object is in a map.

9. (Original) The method of claim 8 further comprising: calling an object; generating an export ID; outputting the JavaScript code to declare the object called as the converter for the object; and adding the ID to the array.

10. (Original) The method of claim 9 further comprising outputting a JavaScript array to hold the

exported IDs.

11. (Previously presented) The method of claim 9 further comprising computing the signature of the object; checking the map for the signature; and if the object is already in the map; retrieving its export ID.

12. (Original) The method of claim 9 further comprising: calling the converter for the object; and outputting the buffer for the exported object values.

13. (Cancelled)

14. (Currently amended) A computer readable medium comprising program instructions for:

a) identifying one or more Java object classes from a web server;

b) determining instance data based on the Java object classes;

c) introspecting each Java object class;

d) automatically creating an artifact representing a software model describing the instance data~~, the artifact mirroring the Java-based object hierarchy~~;

e) generating at least one converter ~~one or more converters~~ for each Java object ~~class~~, ~~each~~ the at least one converter being based on the artifact and configured for receiving the instance data and generating JavaScript code for converting each Java-based object to a JavaScript object and ~~for recreating an instance from the classes as JavaScript objects,~~ for ~~display~~ projecting the JavaScript objects on a browser ~~client of the~~ web ~~server~~page;

wherein the generating step comprises: ~~converters together form a compiled recursive descent parser, synching the object hierarchies by~~

maintaining rules that enforce typing and other ~~from the instance data that enforce Java~~ modeling constraints not found in JavaScript; and

defining a smallest number of properties of each of the JavaScript objects that together represent its uniqueness to produce a string that is portable across any environment and

represents each JavaScript object's uniqueness, such that if there are two physical copies of what is logically the same JavaScript object, the method automatically aliases them out such that in the browser, there is only one copy of each duplicated JavaScript object;

f) generating a simple wrapper for invoking the at least one converter

~~f) invoking the converters at runtime to directly generate JavaScript code for projecting onto the browser client~~.